Replikasi dan Eksperimen Model Information Retrieval Berdasarkan Artikel Jurnal:

Weclustering:Word Embeddings Based Text Clustering Technique for Large Datasets

---

MUKHAMMAD NUR MUZAMMIL (32602100093)
NDARU SENOPATI NUSA (32602100104)
SHENDY CANDRA SUKMA BUANA (32602100114)

Dosen Pengampu:
Badie'ah, ST., M.Kom

Fakultas Teknologi Industri
Universitas Islam Sultan Agung

2025

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

Abstrak

Peningkatan volume informasi berbasis teks menuntut adanya sistem yang mampu mengelompokkan dokumen secara efisien. Salah satu pendekatan yang digunakan dalam bidang *Information Retrieval (IR)* adalah klasterisasi teks. Artikel WEClustering yang menjadi rujukan penelitian ini mengusulkan metode klasterisasi berbasis word embedding (BERT) dan TF-IDF untuk mengatasi kendala polisemi, sinonimi, dan tingginya dimensi dokumen. Penelitian ini bertujuan mereplikasi pendekatan tersebut dengan model yang lebih sederhana menggunakan TF-IDF dan algoritma KMeans, serta membangun aplikasi interaktif berbasis Streamlit. Eksperimen dilakukan pada dataset 20 Newsgroups (subset 2000 dokumen), melalui tahapan preprocessing, vektorisasi TF-IDF, klasterisasi KMeans, dan evaluasi menggunakan Silhouette Score, Adjusted Rand Index (ARI), dan Purity. Hasil menunjukkan bahwa model mampu mengelompokkan dokumen dengan nilai Purity tertinggi (0.8695), meskipun nilai ARI (0.0753) dan Silhouette (0.0383) masih lebih rendah dibandingkan hasil WEClustering. Aplikasi Streamlit yang dibangun berhasil memfasilitasi proses klasterisasi secara efisien dan interaktif.

Kata kunci: Information Retrieval[1], Klasterisasi Teks[2], TF-IDF[3], KMeans[4]

BAB I
PENDAHULUAN

1.1 Latar Belakang

Setiap hari, jumlah informasi berbasis teks yang tersedia secara digital terus bertambah mulai dari artikel berita, forum diskusi, hingga jurnal ilmiah. Tantangan utamanya adalah bagaimana mengelompokkan dan memahami teks-teks tersebut secara efisien. Salah satu pendekatan yang umum digunakan dalam bidang Information Retrieval (IR) adalah klasterisasi dokumen, yaitu proses mengelompokkan dokumen ke dalam kelompok-kelompok yang memiliki kesamaan topik atau isi.

Namun, klasterisasi dokumen memiliki tantangan tersendiri, seperti banyaknya dimensi dari data teks dan makna kata yang bisa berbeda tergantung konteks. Untuk menjawab tantangan tersebut, (Mehta, Bawa, & Singh, 2021) dalam artikelnya memperkenalkan metode WEClustering, yaitu teknik klasterisasi yang menggabungkan kekuatan dari *word embeddings* berbasis BERT dengan pendekatan statistik TF-IDF. Metode ini terbukti menghasilkan klaster yang lebih akurat, terutama pada dataset berukuran besar.

Proyek ini bertujuan untuk mereplikasi eksperimen dari artikel tersebut dalam skala yang lebih sederhana. Kami membangun sebuah aplikasi berbasis Streamlit untuk memproses dan mengelompokkan dokumen menggunakan TF-IDF dan KMeans, serta mengevaluasi hasilnya menggunakan metrik yang sama dengan yang digunakan dalam artikel.

1.2 Rumusan Masalah

Bagaimana performa metode klasterisasi dokumen berbasis TF-IDF dan KMeans pada dataset 20 Newsgroups?

1.3 Tujuan

1 Mereplikasi pendekatan eksperimen dari artikel “WEClustering” menggunakan teknik yang lebih sederhana.

2 Membangun aplikasi interaktif untuk proses klasterisasi teks berbasis TF-IDF dan KMeans.

3 Melakukan evaluasi performa model menggunakan metrik *Silhouette Score, Adjusted Rand Index* (ARI)*, dan Purity*.

4 Membandingkan hasil eksperimen dengan hasil yang dilaporkan oleh artikel utama.

1.4 Manfaat

1 Secara akademis, proyek ini memberikan pemahaman lebih mendalam tentang bagaimana *Information Retrieval* dan *text clustering* diterapkan pada data nyata. Mahasiswa dapat belajar langsung dari penerapan konsep IR dalam sebuah aplikasi interaktif.

2 Secara praktis, aplikasi ini dapat digunakan untuk mengevaluasi performa model klasterisasi teks dan dapat menjadi dasar pengembangan sistem yang lebih kompleks di masa depan.

1.5 Ruang Lingkup

1 Dataset yang digunakan adalah subset dari 20 Newsgroups dengan jumlah dokumen maksimum 2000.

2 Word embedding dalam proyek ini dibatasi pada pendekatan TF-IDF (tidak mencakup BERT karena keterbatasan implementasi).

3 Klasterisasi dilakukan menggunakan algoritma KMeans.

4 Evaluasi terbatas pada tiga metrik: *Silhouette Score*, *Adjusted Rand Index*, dan *Purity*.

BAB II
TINJAUAN PUSTAKA

2.1 Deskripsi Artikel Jurnal yang Dirujuk

Artikel utama yang menjadi dasar proyek ini berjudul "*WEClustering: Word Embeddings Based Text Clustering Technique for Large Datasets*" oleh (Mehta dkk., 2021), yang dipublikasikan dalam jurnal *Complex & Intelligent Systems*. Artikel ini membahas tantangan utama dalam klasterisasi dokumen teks berdimensi tinggi dan menawarkan pendekatan berbasis *word embedding* sebagai solusinya.

Sebagian besar metode klasterisasi teks konvensional, seperti KMeans dan Agglomerative Clustering, masih bergantung pada pendekatan statistik seperti Bag of Words atau TF-IDF. Pendekatan ini hanya menghitung frekuensi kata tanpa memahami konteks maknanya, sehingga tidak mampu menangani persoalan polisemi (satu kata, banyak makna) dan sinonimi (kata berbeda, makna sama). Selain itu, representasi berbasis seluruh kata dalam korpus menyebabkan dimensi data sangat besar (curse of dimensionality) dan bersifat sparsity tinggi, yang menyulitkan proses klasterisasi.

Untuk mengatasi tantangan tersebut, artikel ini mengusulkan metode WEClustering yang menggabungkan representasi semantik dari word embeddings khususnya dari model BERT dengan pendekatan statistik TF-IDF. Tujuannya adalah merepresentasikan dokumen bukan lagi berdasarkan kata individual, melainkan berdasarkan kelompok kata semakna yang lebih ringkas dan bermakna secara kontekstual.

Dataset yang digunakan dalam artikel ini sebanyak tujuh dataset benchmark dengan variasi ukuran dan domain yang berbeda digunakan untuk mengevaluasi seluruh teknik yang diuji dalam penelitian ini. Dataset-dataset tersebut dipilih karena mewakili berbagai jenis data teks yang umum digunakan dalam tugas klasterisasi, seperti berita, forum diskusi, dan dokumen tematik lainnya. Rincian lengkap mengenai masing-masing dataset disajikan pada bagian berikut dan dirangkum dalam table 1:

Table 1 Dataset yang digunakan

No	Dataset	Total Kategori	Total Dokumen
1	Articles-253	5	253
2	Scopus	5	500
3	20NG	4	700
4	Classic4	4	800
5	Scopus-long	7	2800
6	Classic4-long	4	3891
7	20NG-long	9	8131

Secara keseluruhan, metode WEClustering terdiri dari lima tahapan utama, yaitu:

1 *Preprocessing*:

Dokumen diubah menjadi lowercase dan dipisah menjadi kalimat agar sesuai input untuk BERT.

2 Word Embedding

Setiap kata dalam dokumen diubah menjadi vektor berdimensi tinggi menggunakan BERT-large. Embedding ini bersifat kontekstual dan mewakili makna kata dalam kalimat.

3 Clustering Word Embeddings

Vektor kata dikelompokkan menjadi konsep menggunakan MiniBatch KMeans, untuk menghasilkan representasi semantik baru.

4 Concept-Document Matrix

Dibentuk matriks baru yang berisi skor TF-IDF untuk setiap *konsep* dalam dokumen, sebagai pengganti term-document matrix tradisional

5 Document Clustering

Terakhir, dokumen diklasterkan berdasarkan matrix konsep menggunakan KMeans atau Agglomerative Clustering

Untuk membuktikan efektivitasnya, metode WEClustering diuji pada tujuh dataset nyata, salah satunya adalah 20 Newsgroups, yaitu kumpulan dokumen berita dengan berbagai kategori topik. Dataset ini cukup populer digunakan dalam pengujian teknik IR dan klasterisasi karena jumlah dokumennya besar dan memiliki

kategori yang jelas. Selain 20NG, digunakan juga dataset seperti Articles-253, Scopus, dan Classic4. Evaluasi dilakukan menggunakan tiga metrik umum dalam clustering:

1 Silhouette Coefficient

Untuk mengukur seberapa baik suatu dokumen berada dalam klasternya sendiri dibandingkan dengan klaster lain.

2 Adjusted Rand Index (ARI)

3 Untuk membandingkan hasil klaster dengan label asli.

4 Purity

Untuk mengukur seberapa banyak dokumen dalam satu klaster berasal dari kelas yang sama.

Dan untuk performa diringkas dalam Gambar 1 berikut:

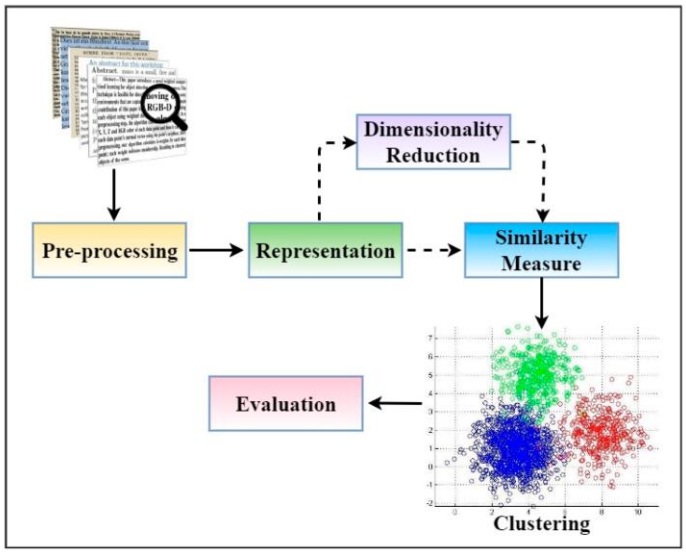
S. no.	Datasets	Min. %age improvement in Silhouette coefficient	Min. %age improvement in Purity	Min. %age improvement in ARI
1.	Articles-253	85.425	0	0
2.	Classic4	3.937	1.029	3.165
3.	Classic4-long	− 35.25	0.828	2.217
4.	Scopus	460.714	4.366	10.656
5.	Scopus-long	410.638	− 2.269	16.603
6.	20NG	306.25	4.942	26.470
7.	20NG-long	244	12.054	68.367

Gambar 1 Performa menggunakan metode WEClustering

2.2 Landasan Teori

A. Information Retrieval

Information Retrieval (IR) atau pengambilan informasi membahas cara menemukan informasi relevan dari kumpulan besar dokumen teks. IR memanfaatkan tiga komponen utama: indexing, similarity measurement, dan ranking untuk menyusun hasil berdasarkan relevansi (Afandi, Homaidi, Ghofur, & Zubairi, 2024)



Gambar 2 Alur Information Retrieval

(https://www.mdpi.com/2076-3417/13/1/342)

Pada Gambar 2, ditunjukkan alur proses Information Retrieval berbasis klasterisasi teks, mulai dari pre-processing, pembentukan representasi dokumen, pengukuran kemiripan, hingga proses clustering dan evaluasi hasil.

Meskipun dalam proyek ini tidak ada query eksplisit, konsep IR tetap penting karena berdasarkan kemiripan dokumen, sistem menghasilkan *clustering* yang juga merupakan bagian teknik IR modern.

B. Indexing

Indexing adalah proses mengubah kumpulan dokumen menjadi struktur pencarian yang cepat dan efisien, biasanya menggunakan inverted index yang menghubungkan term ke dokumennya. Biasanya diawali dengan preprocessing seperti case folding, tokenisasi, stopword removal, dan stemming (Iskandar & Kurniawati, 2025)

Index menjadi dasar representasi dokumen dalam bentuk TF-IDF atau embedding. Dalam klasterisasi teks, pemilihan term dan struktur indeks mempengaruhi akurasi representasi dan efisiensi proses.Representasi Teks: TF-IDF dan Word Embedding

C. Representasi Teks: TF-IDF dan Word Embedding

Ada dua pendekatan representasi dokumen yang umum:

1) TF-IDF (*Term Frequency–Inverse Document Frequency*): Memberikan bobot pada term berdasarkan frekuensi lokal dan global. Contoh: (Al Rasyid & Ningsih, 2024) berhasil menggunakan TF–IDF untuk memberikan hasil pencarian destinasi wisata yang relevan dengan presisi rata-rata 83 %
2) Word Embedding: Merepresentasikan kata dalam bentuk vektor semantik, menangkap makna berdasarkan konteks. Literatur lokal menunjukkan bahwa kombinasi embedding dan TF-IDF memberi peningkatan performa, seperti clustering abstrak ilmiah yang mencapai ARI hingga 0,888.

Pendekatan embedding juga membantu mengatasi isu polisemi dan sinonimi karena memahami konteks kata.

D. Clustering dalam IR

Clustering adalah teknik unsupervised yang digunakan untuk mengelompokkan dokumen berdasarkan kemiripan konten. Algoritma yang sering digunakan:

- KMeans — partisi berdasarkan jarak centroid.
- MiniBatch KMeans — efisien untuk dataset berukuran besar.
- Agglomerative Clustering — metode hierarki yang menyusun cluster secara bertahap.

Evaluasi klaster dilakukan menggunakan:

- Silhouette Score (internal): mengukur kohesi dan pemisahan klaster.
- Adjusted Rand Index (ARI) dan Purity (eksternal): membandingkan hasil klaster dengan label ground truth.

BAB III
METODOLOGI

3.1 Desain Sistem

Metodologi dalam penelitian ini mengikuti tahapan replikasi eksperimen dari artikel "WEClustering" yang disesuaikan dengan implementasi berbasis TF-IDF dan algoritma KMeans. Sistem dibangun dalam bentuk aplikasi interaktif menggunakan Streamlit, yang memungkinkan pengguna mengunggah data, melakukan preprocessing, vektorisasi, klasterisasi, dan mengevaluasi hasil secara langsung. Alur kerja sistem secara umum terdiri dari enam tahap utama:

1) **Preprocessing Teks**

   Dokumen dari dataset dibersihkan melalui proses tokenisasi, case folding, stopword removal, dan stemming.

2) **Ekstraksi Fitur (Representasi Teks)**

   Dokumen yang telah dibersihkan diubah menjadi vektor menggunakan TF-IDF (*Term Frequency–Inverse Document Frequency*).

3) **Klasterisasi Dokumen**

   Vektor TF-IDF digunakan sebagai input ke algoritma KMeans untuk mengelompokkan dokumen berdasarkan kemiripan fitur.

4) **Evaluasi Model**

   Hasil klasterisasi dievaluasi menggunakan tiga metrik: Silhouette Score, Adjusted Rand Index (ARI), dan Purity.

5) **Visualisasi Hasil**

   Hasil klaster divisualisasikan dalam bentuk scatter plot 2D menggunakan metode reduksi dimensi (PCA atau TSNE).

6) **Perbandingan dengan Hasil Jurnal**

   Nilai metrik evaluasi dibandingkan dengan hasil dari artikel WEClustering sebagai acuan keberhasilan replikasi.

3.2 Dataset

Dataset yang digunakan adalah 20 Newsgroups, yaitu kumpulan dokumen teks dari berbagai kategori berita, seperti politik, olahraga, teknologi, dan keagamaan.

Dataset ini terdiri dari ribuan dokumen, namun dalam penelitian ini hanya digunakan subset sebanyak 2000 dokumen untuk efisiensi komputasi.

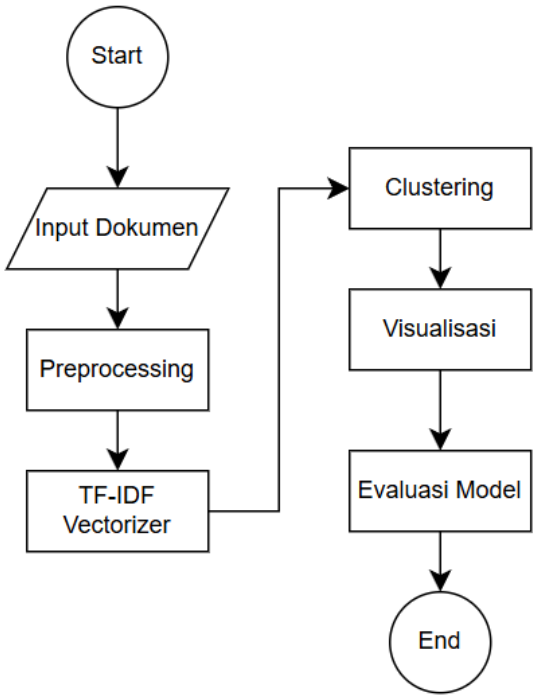
Dataset sudah dalam format terstruktur dan diolah secara lokal, disimpan dalam direktori .tar.gz yang dibaca melalui fungsi pemrosesan Python.

3.3 Implementasi

Aplikasi dibangun menggunakan bahasa Python dengan framework Streamlit. Beberapa library pendukung yang digunakan antara lain:

- scikit-learn untuk TF-IDF, KMeans, dan evaluasi clustering
- matplotlib dan seaborn untuk visualisasi hasil
- pandas untuk manajemen data TSNE atau PCA untuk reduksi dimensi dalam visualisasi

Berikut struktur alur sistem secara teknis:



Gambar 3 *Flowchart* Aplikasi

A. Preprocessing Teks

Tahapan preprocessing dilakukan untuk menyiapkan dokumen menjadi format bersih dan seragam. Proses meliputi:

1. Case Folding: Mengubah semua huruf menjadi lowercase
2. Tokenisasi: Memecah teks menjadi kata-kata

3. Stopword Removal: Menghapus kata-kata umum yang tidak memiliki makna khusus
4. Stemming: Mengembalikan kata ke bentuk dasar

Proses ini dilakukan agar fitur yang diekstrak dari dokumen lebih bersih dan bermakna saat digunakan dalam perhitungan TF-IDF.

B. TF-IDF *Vectorizer*

Setelah dokumen diproses, setiap dokumen diubah menjadi representasi numerik menggunakan TF-IDF. Fitur ini mengukur seberapa penting sebuah kata dalam dokumen dibandingkan dengan seluruh koleksi.

Formula TF-IDF:

$$\text{TF-IDF}(t,d) = \text{TF}(t,d) \times \log\left(\frac{N}{\text{df}(t)}\right)$$

Gambar 4 Rumus TF-IDF

Dimana:

- *TF(t,d):* Frekuensi kata t dalam dokumen d
- *df(t)*: Jumlah dokumen yang mengandung kata t
- *N*: Jumlah total dokumen

C. *Clustering*

Representasi TF-IDF selanjutnya digunakan untuk klasterisasi menggunakan KMeans, yaitu algoritma partisi yang mengelompokkan dokumen berdasarkan jarak ke centroid masing-masing klaster. Nilai k (jumlah klaster) ditentukan dari jumlah kategori dokumen, yaitu 20 klaster.

D. Visualisasi Hasil

Visualisasi dilakukan untuk melihat sebaran klaster dalam ruang dua dimensi. Karena representasi TF-IDF berdimensi tinggi, digunakan metode reduksi dimensi seperti PCA (*Principal Component Analysis*) untuk transformasi linier

Setiap klaster ditampilkan dalam warna berbeda agar pola pemisahan antar klaster dapat terlihat jelas.

E. Evaluasi Model

Tiga metrik evaluasi yang digunakan adalah:

1. Silhouette Score

   Mengukur seberapa dekat dokumen ke dalam klasternya dibanding klister lain. Skor antara -1 (buruk) hingga 1 (baik).

2. Adjusted Rand Index (ARI)

   Mengukur kesesuaian hasil klaster dengan label asli (ground truth). ARI bernilai antara -1 hingga 1; semakin mendekati 1 berarti semakin baik.

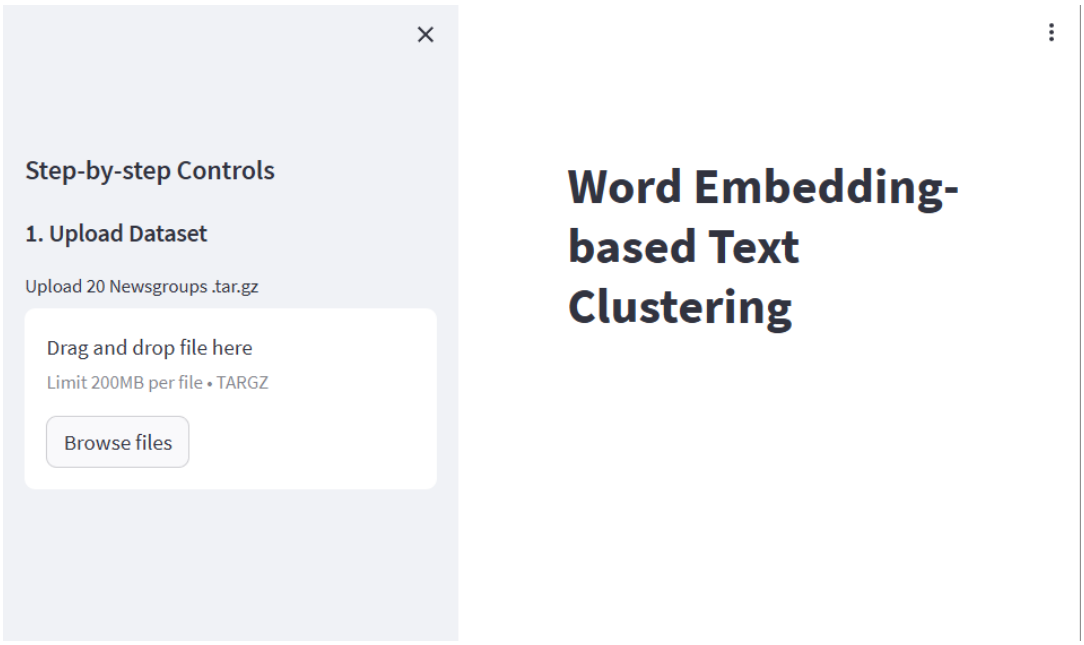
3. Purity

   Mengukur seberapa homogen dokumen dalam setiap klaster. Nilai Purity yang tinggi menandakan bahwa satu klaster didominasi oleh dokumen dengan label yang sama.

BAB IV
HASIL DAN PEMBAHASAN

4.1 Antarmuka Pengguna (UI)

Aplikasi yang dibangun dalam proyek ini menggunakan framework Streamlit, yang memungkinkan pembuatan antarmuka web sederhana dan interaktif langsung dari skrip Python. Antarmuka aplikasi dirancang agar pengguna dapat menjalankan seluruh tahapan klasterisasi teks tanpa harus menulis kode secara manual.



Gambar 5 Halaman Utama Aplikasi

Pada halaman awal, pengguna disuguhkan dengan:

1. Judul aplikasi dan deskripsi singkat.
2. Menu sidebar untuk memilih metode representasi teks (misalnya TF-IDF).
3. Tombol untuk memulai eksperimen atau mengunggah dokumen sendiri.

Gambar 6 Halaman Preprocessing

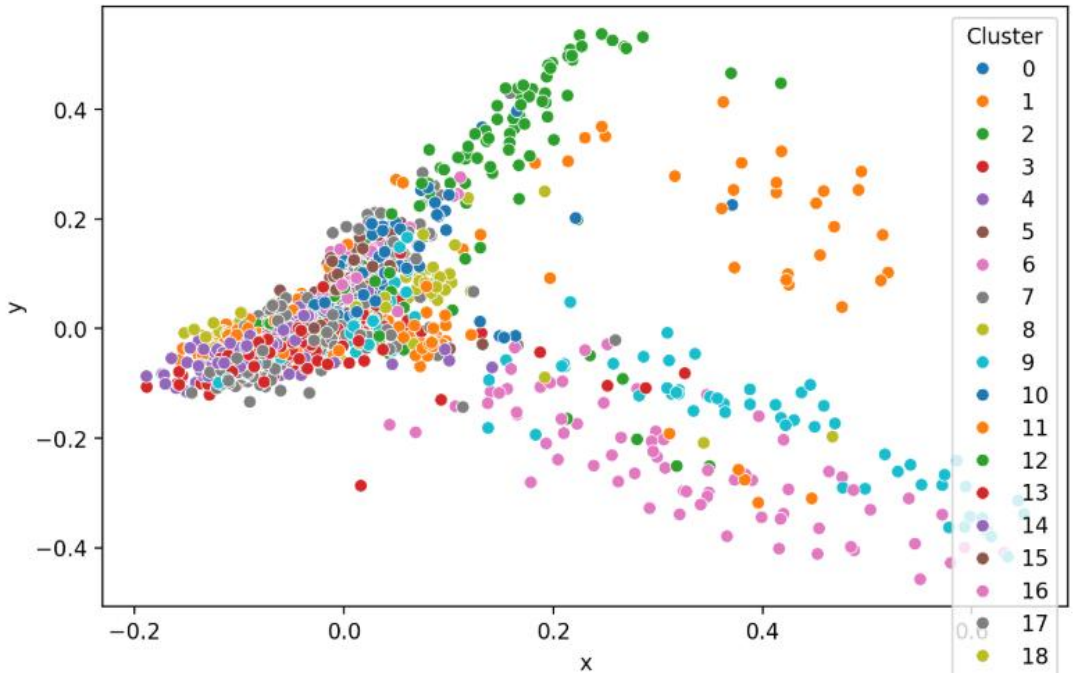
Setelah preprocessing:

1. Aplikasi menampilkan bentuk matriks TF-IDF.
2. Pengguna dapat memilih jumlah klaster
3. Tombol untuk menjalankan KMeans dan menampilkan hasil.

Evaluasi Clustering

	Metric	Score
0	Silhouette	0.0383
1	Adjusted Rand Index	0.0753
2	Purity	0.8695

Visualisasi PCA 2D



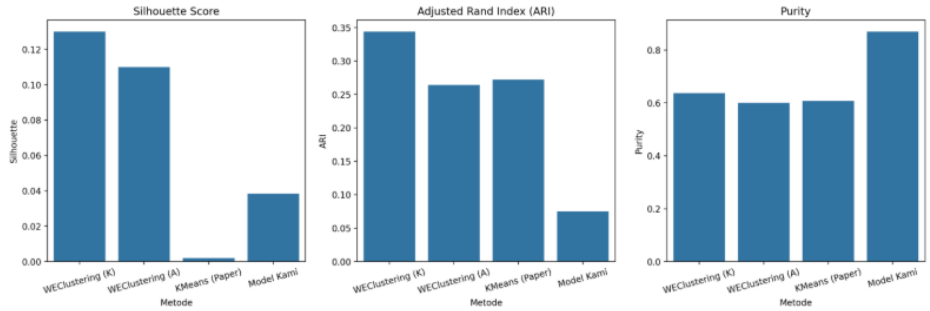
Gambar 7 Hasil Evaluasi Dan Visualisasi

Setelah proses klasterisasi selesai:

1. Ditampilkan tabel metrik evaluasi: Silhouette, ARI, dan Purity.
2. Visualisasi klaster dalam bentuk plot 2D.
3. Tabel klaster yang berisi contoh dokumen dan label aslinya.

	Metode	Silhouette	ARI	Purity
0	WEClustering (K)	0.13	0.344	0.637
1	WEClustering (A)	0.11	0.264	0.6
2	KMeans (Paper)	0.002	0.272	0.607
3	Model Kami	0.0383	0.0753	0.8695

Visualisasi Perbandingan Performa



Gambar 8 Perbandingan Hasil

Gambar 8 diatas yang menampilkan hasil metrik dari eksperimen pengguna dibandingkan dengan hasil yang dilaporkan di artikel WEClustering.

4.2 Perbandingan dengan Hasil Artikel Jurnal

1. Kesamaan

Dalam aplikasi yang dibuat oleh peneliti memiliki kesamaan dengan jurnal yang dirujuk, yaitu:

A. Tujuan dan Dataset yang Sama

Semua metode digunakan untuk mengelompokkan dokumen dalam dataset 20 Newsgroups ke dalam 20 klaster berdasarkan kesamaan topik.

B. Metrik Evaluasi yang Sama

Ketiga metrik yang digunakan untuk menilai kualitas klasterisasi adalah:

- Silhouette Score (kepadatan dan keterpisahan klaster),
- Adjusted Rand Index (ARI) (kesesuaian hasil klaster dengan label asli),
- Purity (dominasi kelas dalam tiap klaster).

C. Algoritma Klasterisasi KMeans Digunakan

Seluruh metode menggunakan KMeans, kecuali satu varian WEClustering yang memakai Agglomerative (WEClustering A)

2. Perbedaan

`

Table 2 Perbedaan Model Kami dan Jurnal yang dirujuk

Aspek	WEClustering (K/A)	KMeans	Model Kami
Representasi fitur	Konsep dari BERT +TF-IDF	TF-IDF Biasa	TF-IDF Biasa
Algoritma	KMeans/Agglomerative	KMeans	KMeans
Silhouette Score	Tertinggi 0.13	Terendah 0.002	Lebih Tinggi 0.0383
ARI	Tertinggi 0.344	Menengah 0.272	Terendah 0.0753
Purity	0.637	0.6	Tertinggi 0.8695

Penjelasannya:

- Silhouette Score: WEClustering K menghasilkan klaster paling terpisah dengan nilai tertinggi (0.13), sedangkan model kami hanya 0.0383. KMeans dari paper menghasilkan nilai sangat rendah (0.002), menandakan klaster yang tidak terbentuk dengan baik.
- ARI (Adjusted Rand Index): WEClustering juga unggul dalam ARI, yang menunjukkan kedekatan hasil klaster dengan label sebenarnya. Model kami memiliki ARI terendah, yang berarti struktur klaster belum mencerminkan kelas aslinya secara baik.
- Purity: Menariknya, model kami justru mencatat nilai Purity tertinggi (0.8695), menunjukkan bahwa dalam satu klaster, mayoritas dokumen berasal dari satu kelas meskipun secara keseluruhan struktur belum ideal (dibuktikan dari ARI rendah).

4.3 Kendala dan Solusi

Selama proses implementasi sistem klasterisasi teks berbasis Streamlit dan replikasi metode dari artikel WEClustering, terdapat beberapa kendala teknis yang dihadapi, berikut penjelasan dan solusinya:

1. **Dimensi fitur TF-IDF yang terlalu besar**
   a. **Masalah**:
      Setelah proses vektorisasi TF-IDF, jumlah fitur (kata) yang dihasilkan sangat besar, mencapai ribuan. Hal ini menyebabkan beban komputasi meningkat saat proses klasterisasi dan evaluasi.
   b. **Solusi**:
      Dilakukan pembatasan jumlah fitur menggunakan parameter

max_features saat inisialisasi TfidfVectorizer. Selain itu, hanya digunakan subset 2000 dokumen dari dataset untuk menjaga performa tetap stabil di perangkat lokal.

2. **Visualisasi Clustering kurang terlihat jelas**
   a. **Masalah:**
      Visualisasi hasil klaster (dalam bentuk scatter plot 2D) tidak menunjukkan pemisahan klaster yang jelas, karena data asli berdimensi tinggi (ribuan dimensi).
   b. **Solusi**:
      Diterapkan metode PCA (*Principal Component Analysis*) dan T-SNE untuk mereduksi dimensi data agar bisa divisualisasikan ke dalam 2 dimensi dengan pola pemisahan yang lebih terbaca.
3. **Tidak Tersedianya Word Embedding (BERT) dalam Aplikasi**
   a. **Masalah**:
      Artikel WEClustering menggunakan representasi word embedding dari BERT, namun model kami belum mengimplementasikan embedding karena keterbatasan sumber daya.
   b. **Solusi**:
      Sistem difokuskan pada penggunaan TF-IDF sebagai baseline yang tetap relevan untuk tugas klasterisasi dasar. Rencana pengembangan lanjutan mencakup integrasi BERT dengan pemrosesan teks via `transformers` dan GPU runtime.
4. **Evaluasi ARI dan Purity Membutuhkan Label Asli**
   a. **Masalah**:
      Metrik ARI dan Purity membutuhkan label ground truth, padahal sistem IR umumnya tidak menggunakan label (unsupervised).
   b. **Solusi**:
      Sistem diuji menggunakan dataset 20 Newsgroups yang memiliki label, sehingga metrik tetap bisa dihitung untuk keperluan eksperimen. Untuk aplikasi di dunia nyata tanpa label, bisa difokuskan pada metrik internal seperti Silhouette Score.

BAB V
KESIMPULAN

5.1 Kesimpulan

Penelitian ini bertujuan untuk mereplikasi pendekatan klasterisasi teks dari artikel WEClustering dalam bentuk aplikasi interaktif menggunakan Streamlit. Proses klasterisasi dilakukan dengan pendekatan berbasis TF-IDF dan algoritma KMeans, serta dievaluasi menggunakan tiga metrik utama, yaitu Silhouette Score, Adjusted Rand Index (ARI), dan Purity.

Hasil eksperimen menunjukkan bahwa sistem yang dibangun mampu mengelompokkan dokumen ke dalam klaster-topik yang cukup baik. Nilai Purity yang dicapai adalah 0,8695, yang menunjukkan bahwa sebagian besar dokumen dalam suatu klaster berasal dari kelas/topik yang sama. Hal ini mengindikasikan bahwa metode TF-IDF masih cukup efektif dalam mengenali kata-kata kunci utama dari dokumen.

Secara umum, sistem yang dibangun telah berhasil mereplikasi proses klasterisasi dengan cukup baik sebagai baseline sederhana. Hasil evaluasi juga dapat menjadi landasan untuk pengembangan sistem Information Retrieval yang lebih lanjut menggunakan representasi semantik dan algoritma yang lebih kompleks

5.2 Saran

Sebagai pengembangan lebih lanjut, berikut beberapa saran:

1. Mengintegrasikan representasi berbasis word embedding seperti Word2Vec atau BERT untuk meningkatkan kemampuan semantik dalam klasterisasi.
2. Mencoba algoritma alternatif seperti DBSCAN, Agglomerative Clustering, atau MiniBatch KMeans untuk membandingkan performa.
3. Menambahkan evaluasi tambahan seperti Davies–Bouldin Index atau Calinski–Harabasz untuk memperkuat analisis.
4. Menggunakan dataset yang lebih besar dan beragam agar sistem dapat diuji dalam skenario dunia nyata.
5. Menambahkan fitur unggah dataset dan ekspor hasil evaluasi ke format laporan dalam aplikasi Streamlit.

DAFTAR PUSTAKA

Afandi, Muhammad Dzikry, Homaidi, Ahmad, Ghofur, Abd, & Zubairi, Ach. (2024). Penerapan Information Retrieval dalam Sistem Analisis Kemiripan Proposal Skripsi menggunakan Cosine Similarity. *Swabumi*, *12*(1), 39–46. https://doi.org/10.31294/swabumi.v12i1.17087

Al Rasyid, Rio, & Ningsih, Dewi Handayani Untari. (2024). Penerapan Algoritma TF-IDF dan Cosine Similarity untuk Query Pencarian Pada Dataset Destinasi Wisata. *Jurnal JTIK (Jurnal Teknologi Informasi dan Komunikasi)*, *8*(1), 170–178. https://doi.org/10.35870/jtik.v8i1.1416

Iskandar, Dede, & Kurniawati, Ana. (2025). Analisis Perbandingan Teknik Word2vec dan Doc2vec dalam Mengukur Kemiripan Dokumen Menggunakan Cosine Similarity. *Jurnal Teknologi Informasi dan Ilmu Komputer*, *12*(1), 133–144. https://doi.org/10.25126/jtiik.2025129143

Mehta, Vivek, Bawa, Seema, & Singh, Jasmeet. (2021). WEClustering: word embeddings based text clustering technique for large datasets. *Complex and Intelligent Systems*, *7*(6), 3211–3224. https://doi.org/10.1007/s40747-021-00512-9

Lampiran

```python
@st.cache_data
def purity_score(y_true, y_pred):
    matrix = confusion_matrix(y_true, y_pred)
    return np.sum(np.amax(matrix, axis=0)) / np.sum(matrix)

if uploaded_file:
    with st.spinner("Mengekstrak dataset..."):
        extract_path = extract_dataset(uploaded_file)

    max_docs = st.sidebar.slider("2. Jumlah Dokumen yang Digunakan", 100, 2000, 1000, step=100)
    documents, labels_true = load_documents(extract_path, max_docs=max_docs)
    st.success(f"{len(documents)} dokumen berhasil dimuat.")

    if st.checkbox("Tampilkan cuplikan dokumen"):
        st.subheader("Cuplikan Dataset")
        st.dataframe(pd.DataFrame({"Topik": labels_true[:5], "Dokumen": documents[:5]}))

    st.sidebar.subheader("3. TF-IDF Vectorization")
    max_features = st.sidebar.slider("Jumlah fitur TF-IDF", 500, 5000, 1000, step=500)
    vectorizer = TfidfVectorizer(max_features=max_features, stop_words='english')
    X = vectorizer.fit_transform(documents)

    st.sidebar.subheader("4. Clustering")
    k = st.sidebar.slider("Jumlah Cluster (KMeans)", 2, 30, 5)
    run_cluster = st.sidebar.button("Jalankan Clustering")

    if run_cluster:
        with st.spinner("Melakukan clustering..."):
```

Name	Date modified	Type	Size
A long time ago			
comp.graphics	04/04/1999 01:14	File folder	
talk.politics.guns	04/03/1997 02:43	File folder	
talk.politics.mideast	04/03/1997 02:43	File folder	
talk.politics.misc	04/03/1997 02:43	File folder	
talk.religion.misc	04/03/1997 02:43	File folder	
comp.sys.mac.hardware	04/03/1997 02:43	File folder	
comp.windows.x	04/03/1997 02:43	File folder	
misc.forsale	04/03/1997 02:43	File folder	
rec.autos	04/03/1997 02:43	File folder	
rec.motorcycles	04/03/1997 02:43	File folder	
rec.sport.baseball	04/03/1997 02:43	File folder	
rec.sport.hockey	04/03/1997 02:43	File folder	
sci.crypt	04/03/1997 02:43	File folder	
sci.electronics	04/03/1997 02:43	File folder	
sci.med	04/03/1997 02:43	File folder	
sci.space	04/03/1997 02:43	File folder	
soc.religion.christian	04/03/1997 02:43	File folder	
comp.os.ms-windows.misc	04/03/1997 02:43	File folder	

Evaluasi Clustering

	Metric	Score
0	Silhouette	0.0383
1	Adjusted Rand Index	0.0753
2	Purity	0.8695

Visualisasi PCA 2D